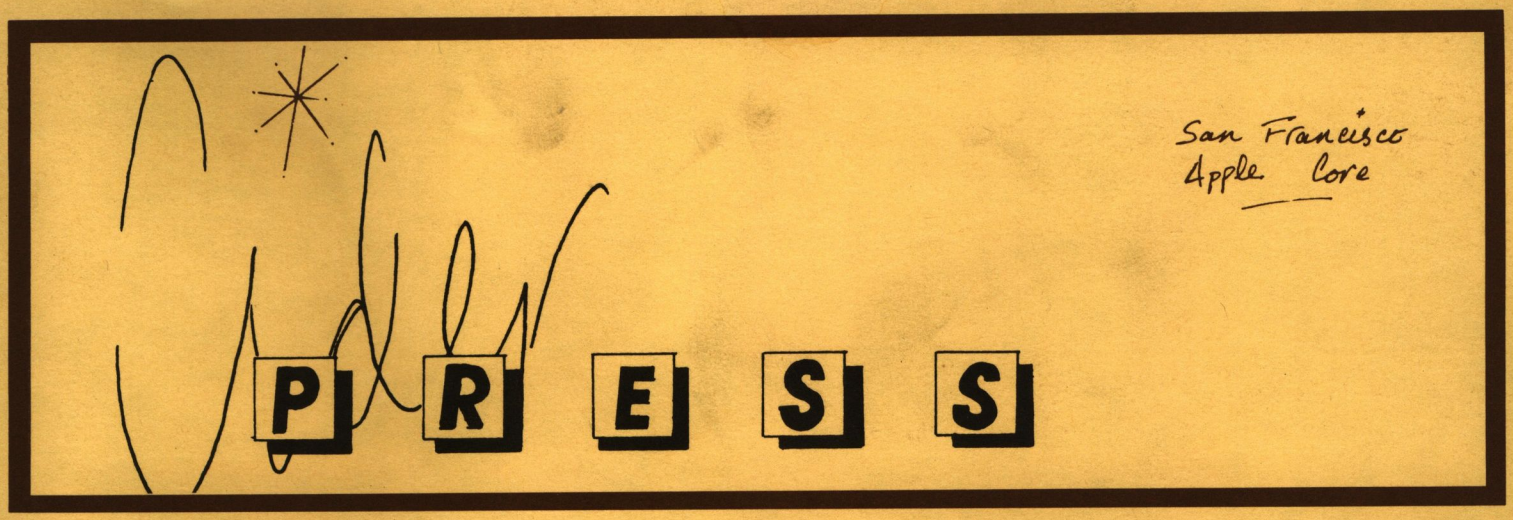


San Francisco Apple Core

IN THIS ISSUE

REM 1
Letters 2
July Meeting Notes - Bill Neff 3
August Coming - Michael Fischer 3
Perfect Numbers - Ed Bronstein & David Frick. 4
Poke 104,96 Did You? - Reed Harris 5
Breakthrough! - Pat Buck 7
Call With Parameters - Reed Harris 8
Epson Salts - Part 3 - Michael Fischer 5
B.A.S.I.C. - Joyce Conklin 12
Pascalia & Fugue
Addressing Envelopes - Max Nareff 14
Pascal Book Review - Max Nareff 12
Swap Meet! 15
Aztec C - a Review - David B. Anderson 16
Marketing Notes - Don Scellato 18
Contest Notes - Don Scellato 18
Adopt-A-School 18
Telecommunications - Frank Fennelley 19
VisiCalc - Don Scellato 19
Interest Groups - Sami Capron 20
Computer Camps - Sami Capron 20
Financial Statement 20
Calendar 21
Club Offerings 22

GENERAL MEETINGS:
AUG 20
SWAP MEET! SEP 10

AUGUST, 1983

CIDER PRESS

REM It occurred to me again that the exchange of information between members is what makes our association one of the best and most valuable resources available to all of us in close intimate contact with these machines which sometimes interfere with our close intimate contacts.

There are worlds behind the keyboard and through the monitor window. No one person has mapped all the territories within, yet each of us has been somewhere in there. We have our favourite haunts where we cavort with control codes, dance with pixels or weave our way through menu-mazes.

Occasionally, we are given the opportunity to travel with a friend; to guide someone through a portion of the realm more familiar to us and less familiar to them. These exchanges are at the heart of it all. They are what the **Apple Core** is all about; they are why we publish the ***Cider Press.***

Avail yourself of all these things by offering to lead the way or by asking directions. The journey will be pleasant and the rewards manyfold.

After a grueling game of Astro-Shoot, I hit the Space Bar.

- ed.

THE CORE'S CORE

DIRECTORS:

Sami Capron	415/388-8631
Joyce Conklin	415/341-6955
Michael Fischer	415/933-4435
Gary Knox	415/521-0603
Stephen C. Lloyd	415/571-7370
Don Scellato	415/672-2051
Alan L. Smerican	415/566-2342
Allan B. Woods	415/687-9659

MEMBER SERVICE/COMMITTEES:

Membership		
Alan L. Smerican		415/566-2342
	Ted Chatterton	
	Sami Capron	
Cider Press -		
Editor	Allan Woods	415/687-9659
BASIC	Joyce Conklin	415/341-6955
Int. Grp. Coord.	Sami Capron	415/388-8631
Marketing Dir.	Don Scellato	415/672-2051
Review Group	("Woody") Liswood	415/585-6375
General Meeting	Michael Fischer	415/933-4435
Exec. Secretary	Stephen C. Lloyd	415/571-7370
Treasurer	Gary Knox	415/521-0603
Coordinating Dir.	Joyce Conklin	415/341-6955

CIDER PRESS POLICY

All manuscripts, photographs, and other materials will be considered submitted free and released for publication. They become the property of the San Francisco Apple Core and the CIDER PRESS. Authors should clearly mark all material submitted for publication so that proper credit can be given.

The views expressed in any articles are those of the authors, and not necessarily those of the San Francisco Apple Core, Inc., the Editor, Club Officers or Directors . . . or anyone else in their right minds.

The San Francisco Apple Core Incorporated is a non - profit organization comprised of and supported by owners and users of Apple I I and Apple / / / computers. The Club is run by officers and committees, and the Club endeavors to aid Apple owners through its Program Libraries, Program Disks, meetings, and this publication.

Article Submission: The CIDER PRESS welcomes any and all articles dealing with Apple computers and their associated hardware and software. Articles may be submitted (typed or printed, double - space), or as disk text files using any of the popular word processors or text editors. Any author who has an article published in the CIDER PRESS will receive a free copy of the most current Disk - of - the - Month.

Reprints: All articles appearing in the CIDER PRESS not copyrighted by the author may be reprinted by other *non - profit* Apple User Groups, without prior permission, so long as proper credit is given to both the author and the CIDER PRESS. Permission to reprint a copyrighted article must be obtained in advance, by writing to the authors, c/o the San Francisco Apple Core.

San Francisco Apple Core
1515 Sloat Blvd.
San Francisco, CA 94132

Letters

Pascal programmers who write programs that are constantly threatening to fall off the 64k edge and onto the floor may be interested to know that certain 80-column cards take up more memory than others. Or perhaps certain 80-column cards free up more memory that others.

I've run the following program on an Apple II+ and various 80-column cards:

```
PROGRAM TEST;
BEGIN
  WRITELN(MEMAVAIL);
END.
```

80-column card	mem. avail.
none	18824 words
Videoterm (Videx)	18824
Smarterm II (ALS)	19336
Viewmax-80 (Micromax)	19336

Obviously the Smarterm and the Viewmax free up 512 words. Perhaps someone could explain this mystery and answer the next logical question: Is there any more memory where that came from? Do other configurations have other amounts?

The Apple II that I used on the above tests just had the standard two floppies, language card and Epson-clone printer. An Apple II that had a hard disk on it had a number higher than 19336, but I forget what it was.

This could be an interesting diversion while we're waiting for Apple to come out with the 128k version of Pascal.

John Caron
40 Steiner
S.F. CA 94117
863-8474

▪▪▪▪▪▪▪▪▪▪▪▪

Dear Editor,

I am writing to add a little to the article by Dr. Max Nareff, in the June issue of the *Cider Press*.

In his excellent article he gives a tip for using the Pascal filer as a typewriter to do simple tasks such as mailing labels.

Being hooked on Pascal I immediately booted up my system which consists of an Apple II+, Videx 80 column card, language card, Grappler+, bufferboard and disk controller card.

When in the f)iler, I typed console:, printer: and typed and got nothing. You will need to do error-less typing and when finished type a control C and then you will get the message, console: ------> printer:

Then this neat technique will work. This may be just indigenous to my particular machine, but if not try it on yours.

I hope this will help someone. Tell Max to keep it up -- his articles are great.

Sincerely yours,
Michael B. Rumelt, M.D.
Creve Coeur, MO

▪▪▪▪▪▪▪▪▪▪▪▪

Thanks to Michael for reminding us that printers, like people, differ. The Apple Pascal Operating System Manual, pg. 38, confirms this, stating that "some printers require you to press the RETURN key before pressing CTRL-C. Then all your typing will be sent to the printer." Others, transfer input merely with a CTRL-C.

In the short Sept./Oct. 1982 issue of the Journal of Pascal & ADA, a short and ingenious typewriter program was presented. In essence, it is a direct keyboard to print-head transfer, just like a TYPEWRITER. We are attempting to obtain permission to reprint this algorithm. If there are any other printing hints out there, please send them along.

See you soon,
Max Nareff.

Meeting Notes

July Past

-Bill Neff

July 9, 1983
Ft. Mason

Well, it's good old summertime, so a whole bunch of us were out laying in the sun or something like that, because there weren't quite 100 members, officers, kiddies and oldtimers.

After Mike review of the general format, Don introduced and thanked several of last years board of governors: Ken Silverman, Woody Liswood, and Rance Fields. Actually, these members have been very active in club affairs over the past 5 years.

Don also announced prize winners and reminded us of the continuing contests. If your Apple Core utility disc bombs, bring it back for a replacement. We'll have a booth at the Oakland swap meet. Call Don to work in it.

Ken answered several questions about letter quality printers including the new inexpensive typewriter based machines. Next meeting will be Aug 20 and will feature Walt Broder. Sept 10 is our swap meet.

David Green took us on tour through the intricacies of Think Tank, advertised as the idea processor. Actually it is an extremely versatile, menu drive line or outline manipulator with text option and a whole flock of output options. The options on manipulation seem infinite. You have to see it to believe. It seems worth its price of $150. (goodbye scissors and paste)

Phil Prasek didn't have to spend much time showing us his SXR+ cross reference lister. It is very simple, obvious, and fast. Where is what in your Applesoft program? It will list out all the line numbers containing whatever: command, string, variable, constant, parameter, whatelse. A very definite need for editing your programs and text files. Now watch this price! $29 for this version.

The new version which comes out in November with more options and optional at extra cost co-resident utilities will run (briefly) for $35 (or your old version and $6). Send in the coupon with $7.50 and you get another complete kit!

Joyce Conklin presented a professional filing system called Datafax.

— — — —

August Coming

-Michael Fischer

The meeting for August 20 will feature Walter S. Broeder, the person who designed the Apple //e. (Those of you with //e's can look on the motherboard for his initials. He is presently marketing an RGB interface to the Apple.

Mr. Broeder is one of the leading experts on Apple video and will be speaking on that subject. So if you want to know what RGB means and how NTSC works (or what it means) and how all this affects the picture that the Apple displays, come to our Saturday, August 20 meeting beginning at 10:00 a.m. in Room 100, Building C, Fort Mason, San Francisco.

Also at the August meeting Don Scellato, our marketing director, will demonstrate Supertext 40-52-70, a word processor from Muse software that displays 70 characters across the screen without the need for any hardware adapters.

We will also have the sales table, general questions and answers, kids corner, and who knows what else.

Swapmeet

The swap meet is scheduled for our September meeting. The date is one week earlier than usual -- September 10. See the article on the swap meet elsewhere in this issue.

Upcoming . . . an interest group extravaganza, games fest, more speakers and demos. Any ideas? Call Michael Fischer (557-1250 days or 933-4435 evenings).

Perfect Numbers

-Ed Bronstein & David Frick

The June, 1983 issue of the *Cider Press* contained an assignment by Joyce Conklin to program in BASIC a Perfect Number finder. I'd never heard of perfect numbers before and I can't program in BASIC but the problem sounded interesting so I wrote something in FORTH to try to find a solution.

After testing many thousands of numbers, only four showed up as perfect: 6, 28, 496, 8128. These numbers seemed to have some interesting properties: all factors of these numbers are even except for one. Later I found that the one odd number must be prime. Further, the factors bear a strange relationship to the way in which computers count. Do the numbers 16, 32, 64 sound familiar?

I mentioned perfect numbers to a computer - genius - mathematician friend, David Frick of Frick Laboratories, Atherton. He sat down and jotted off a number finder in BASIC on his heath and analyzed results. He grew very excited when he came up with a formula for finding perfect numbers:

$$2^{(n-1)} * (2^{(n)}-1)$$

In other words, if n is equal to 3, then 2 to the power of 3-1 (2 squared = 4) times 2 cubed-1 (=7) gives a product of 28 - - a Perfect Number. A requirement: the second term must be prime!

Later Dave called to say that he had gone back to some of his old math books and it seemed that someone else had come up with the same formula previously -- a fellow named Euclid who messed around with numbers a few years ago in Greece.

Frick's BASIC program accompanies. It runs extremely fast because he used some very clever short cuts. For instance, no factor is examined that is greater than the square root of the number. Example: 4 is less than the square root of 28 and is evaluated, but there is no need to test 7 because it is part of 4's factoring into 28. Also, there is no need to test remaining numbers which are powers of 2: if 4 is a factor of 28, then 2 is a factor as well.

Credit for the square root trick goes to another Greek citizen, Aristophanes of Byzantium. The technique is known as the Sieve of Aristophanes.

Other perfect numbers are: 33,550,336; 8,589,869,056; 137,438,691,328 and on and on. The last two were not tested. Would somebody like to try the following number?

$$2^{2280}*(2^{2281}-1)$$

Is the second term prime?

PERFECT NUMBER SEARCHER
by David Frick

Enter an even number and the next perfect number will be found.
Enter an odd number and prime numbers will be cranked out forever.

```
2 N9 = 2: REM  STEP OF #'S TO TEST

9  DIM A(1000)

13  INPUT NO

14 NO = NO - N9

15  PRINT

16 NO = NO + N9

18  PRINT NO;

21  PRINT "=";

22 I = 1

23 A(1) = NO: IF NO = 0 THEN 15

25 N = A(I)

30  GOSUB 1000

40 A(I) = F:I = I + 1:A(I) = T

75  IF T <  > 1 THEN 25

400 I = I - 1

410  FOR J = 1 TO I

420  PRINT A(J);"*";

430  NEXT J

432  PRINT "1";

450  IF I = 1 THEN 900

500  GOSUB 2000
```

```
505  IF YO = 0 THEN 520

510  PRINT " ///";YO;

520  IF YO <  > NO THEN 15

530  PRINT : FOR Z = 1 TO 10: PRINT
   "***PERFECT***"; CHR$ (7);: PRINT NO:
   NEXT : END

540  PRINT NO;

600  PRINT

605  FOR I = 1 TO 3

610  PRINT  CHR$ (7);

615  FOR J = 1 TO 200

616  NEXT J

620  NEXT I

630  GOTO 530

800  GOTO 15

900  PRINT "---PRIME---"; CHR$ (7);

910  GOTO 15

1000  IF N = 2 *  INT (N / 2) THEN F =
   2:T = N / 2: RETURN

1005 U =  INT ( SQR (N))

1010 F = 1

1020 F = F + 2

1030  IF F > U THEN F = N:T = 1: RETURN

1040 T =  INT (N / F)

1050  IF N <  > F * T THEN 1020

1060  RETURN

2000  REM  CHECK FOR PERFECT

2010 I1 = I + 2

2015  FOR N = 1 TO I

2017  PRINT "+";

2020  FOR J = 1 TO I

2050 X = 1

2100  FOR K = 1 TO N

2110 Y = A(J + K - 1)

2120  IF Y = 1 THEN 2220

2130 X = X * Y

2140  NEXT K

2145  IF A(I1 - 1) = X THEN 2210

2200 A(I1) = X:I1 = I1 + 1

2205  PRINT "(";X;")";

2210  NEXT J

2220  NEXT N

2300 YO = 1 - NO

2305 A(I1) = 0

2310  FOR J = I + 2 TO I1 - 1

2315  GOTO 2350

2320  FOR K = J + 1 TO I1

2330  IF A(J) = A(K) THEN 2400

2340  NEXT K

2350 YO = YO + A(J)

2360  IF YO > NO THEN  PRINT YO;:YO = 0:
   GOTO 3000

2400  NEXT J

2999  RETURN

3000  PRINT " >>> ";NO

3020  RETURN
```

▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪

POKE 104,96 DID YOU ??

(c) 1983 by Reed Harris

So you typed in "POKE 104,96: POKE 103,1: POKE 24576,0", return, "10 TEXT" (or any other BASIC statement), return, and you got a screen full of garbage. Not only that, you lost control of your computer. ?? how and why ??

Locations 103(1) and 104(h) contain the start-of-BASIC-program address. When you turn on your Apple these locations contain 01(1) and 08(h) = $0801 in hex (or 2049 in decimal).

Two other locations, 175(1) and 176(h) contain the end-of-program address. Initially they contain 03(1) and 08(h) = $0803 hex (or 2051 dec).

When you poked 104,96 you moved the start of program pointer way above the end of program pointer. The program ended before it began so to speak and

the 6502 couldn't make sense of it so it left you out of control.

At least that's my explanation of it. I do know that typing "NEW", return, before entering any BASIC statements will prevent the problem. Also, "NEW" changes the end-of-program address to equal the start-of-program address plus 2 and inserts the two bytes of zeros which are the end-of-program marker. Those two zero bytes would otherwise be the address of the start of the next BASIC statement.

Incidentally, if you run a program and then it won't list, go into the monitor (CALL-151) and look at "800.860". Maybe they put zeros in $801 and $802 to stop you from listing it (or maybe you typed "new" before thinking and now want to recover the program).

Look for a sequence of five bytes like "00 aa bb nn mm". The first byte is always zero and is the end marker of the previous statement. The next two bytes, aa bb, are the address, lo hi, of the next statement.

The fourth and fifth bytes, nn mm, are the line number of this line which follows and which ends with "00". The "aa" is the lo byte of the address of the next statement and its address is contained in the first two bytes of this statement. The "mm" is zero when the line number is less than 256 dec).

Look at the example below. With 00 in $801,802 the program won't list. $0800 is the first "00", the first set of five bytes so look for the next one. Here we find it at $0808. at $0809.080A, we find "39 08". The value $0839 is the address of the third statement and is the starting place of the second. type "801: 09 08", return, because $0809 is the address of the start of the second statement and we need that address at the start of the first statement.

Now you can list it from Applesoft. Don't forget that in the monitor all the numbers are hex. There is a good hex to dec table in 'Best of Cider Press 78-79' by Rod Carlisle.

```
0800- 00 00 00 05 00 89 3A 97
0808- 00 39 08 0A 00 BA 22 54
0810- 48 49 53 20 49 53 20 41
0818- 20 53 48 4F 52 54 20 50
0820- 52 4F 47 52 41 4D 22 3A
0828- 81 4B D0 30 C1 35 30 30
0830- 3A 4B D0 4B C8 31 3A 82
0838- 00 4C 08 14 00 91 3A 93
0840- 30 2C 30 C1 31 30 30 2C
0848- 31 35 30 00 52 08 2C 01
0850- 80 00 00 00 xx xx xx xx
```

```
5   TEXT : HOME
10  PRINT "This is a short program"
          :FOR K = 0 TO 500 : K=K+1:NEXT
20  HGR : HPLOT 0,0 TO 100,150
300 END
```

This item is brought to you by Reed Harris who has learned a lot in the East Bay Self Help Group. I hope it is useful to you.

FOR YOUR APPLE**

80/132 COLUMN CARD $99*/$119*

Expand your APPLE with 132 COLUMN display (20MHZ Monitor - Need for 132 MODE only)
Card comes standard for 80 COLUMN display with an optional 132 COLUMN capacity.
True desenders and compatible with VISICALC** with a preboot diskette. If card ordered with the 132 COLUMN option $119.00.
Basic 80 COLUMN $99.00. 132 COLUMN option can be ordered later for $29.50. Preboot for VISICALC** add $24.50. Shipping date 9/1/83.

II**e 64K 80 COL CARD $99*

Expand your IIe** with 80 COLUMNS and 64K MEMORY

e-Z CARD $60*

Uses Z80A CPU (Needed for WORDSTAR**, D BASE II**, MULTIPLAN** and runs all 2.2 CPM** PROGRAMS
Software not provided.

WISHCARD $50*

Orbital Systems, Inc.

For making backup's of your UNCOPYABLE PROGRAM DISK.
A MUST FOR GAME PLAYERS. Save different levels of games to achieve higher scores
or modify programs in COPYA FORMAT. Utility software included.

*ADD $30.00 IF YOU WANT A READY TO USE UNIT
ALL cards come in KIT form and have ALL REQUIRED IC's, Printed Circuit Board, and Assembly Instructions.

GUARANTEE IF YOU ARE UNABLE TO GET YOUR KIT TO WORK, for a fee of $30.00 plus $2.50 postage and handling (for socketed boards only, not provided), we will correct any errors in soldering and or blown chips, and mail your card back ready to use.

ORBITAL SYSTEMS INC.
2929 E. JASMINE ST. MESA, ARIZONA 85203

FOR ORDER PLACING ONLY 1-800-221-9280 EXT.990$ F 24 HR TOLL FREE 7 DAYS A WEEK
ARIZONA RESIDENTS CALL COLLECT FOR ORDERS 602-830-6457 9 A.M. TO 3 P.M. MONDAY THRU FRIDAY

MASTER CHARGE AND VISA ACCEPTED
Arizona residents add 5% for sales tax

Dealers Inquiries Invited

**CPM trademark Digital Research
**Multiplan trademark of Microsoft
**D Base II trademark of Ashton Tate
**Apple is the trademark of Apple Computer Inc.
**Wordstar is a trademark of Micro Pro Inc.
**Visicalc is a trademark of Visicalc Corp.

BREAKTHROUGH!

by

Pat Buck

I'm writing this in response to a plea for articles in the *Cider Press* and in deep gratitude to Michael Fischer who spent 5 minutes of his time and opened up worlds for me! Here's the story.

I've had my computer about a year and a half and bought my dot matrix printer, the NEC 8023 last year at the West Coast Computer Faire. The documentation that came with it seemed as if written in Japanese, translated into Sanskrit, which was then transcribed into English by a happy drunk whose native language was Hungarian. After weeks of trying to figure out the printer, I could only get it to print normally, but with the Grappler interface documentation, I could get it to print graphics. I knew there was far more the printer was capable of doing.

I phoned the place where I had bought the printer (very helpful people, by the way) who informed me that the documentation that came with the printer was for the TRS-80 computer and not for the Apple. Great! Furthermore, there was no documentation for Apple users at that time, however, this printer was sufficiently like the C-Itoh printer that most of the commands would work on the 8023. So far, progress! They sent me a Xerox copy of this. This, in one fell swoop, gave me too much information and not enough information.

I have always prided myself on being a relatively intelligent individual, and I like the challenge of reading directions and being able to follow them successfully. Thus, knitting directions and sewing directions, even from Vogue, were easy as pie. I was finally, after hours of study and experimentation, able to print different sizes and intensities directly from the keyboard.

Now, the plot thickens. I do a lot of writing, both in my job and for pleasure. I was and remain intrigued by the many word processing programs available to the Apple owner. I have also upgraded my computer with the Videx card, 16 K RAM card (and lately the Z-80 card). Six months ago I bought the AppleWriter][which I like very much and use often. The built-in underline command does not work with my printer. I bought GlossaryOne at one of the Core meetings for my school which has an MX-80 and tried those commands on my printer in hopes of them working. No luck. And underline looked like t-h-i-s. Foiled again!

My ambition was to buy WordStar. Although I have several really great word processing programs (Bank Street Writer is great for dashing off something simple) I had always heard that WordStar was the state of the art word processor. So I bought it at this year's West Coast Computer Faire, but put it away and out of sight until the summer when I would have more time to play with it.

Summer came, out came WordStar. Speaking of incomprehensible, have you ever looked at that documentation? I literally spent three weeks reading and rereading that book. I knew the program did all the wonderful printing things and my printer could do most of them, but I couldn't get them to connect. I asked several people I know at different computer stores, and they suddenly looked dazed, and their eyes wandered off to the horizon. I even phoned MicroPro to ask for a hint, I was given a recorded message telling me to contact my dealer.

I began to attack the problem by a process of elimination hoping that I would stumble on the right track. I only succeeded in having my text start in the middle of the page and overprint half of the next line. I was reduced to whimpering frustratedly in front of my computer.

Which brings me nearly to the end of my story. At the July Apple Core meeting, I asked if <u>anyone</u> could give me some clue how to install my NEC 8023 printer with WordStar. Michael Fischer spent 5 minutes with me going over his article on page 4 and 5 of the July *Cider Press.*

It lit the electric light bulb in my head, and I went home to try it out. It meant about 1/2 hour of research in my printer's codes, but I followed Mr. Fischer's direction, and

IT WORKED !

But the true end of this story is that I was so inspired that I researched my AppleWriter][directions and came upon the CTRL V command, [v], which can be found in one tiny paragraph on page 51. Using this command I can embed my printer commands in the text using AppleWriter][and have my printer do all the goodies I want it to do. This article is written using AppleWriter][.* Now I am wondering if all along I wasn't the only person in the world who

didn't know these tricks, it seems so simple. I tried to put all the commands in a Glossary file, but that didn't work. Any suggestions?

For any of you who want to try it, here are the commands I embed for the NEC 8023 printer.

to underline	[v] ESC SHIFT X [v]
to stop underline	[v] ESC SHIFT Y [v]
to begin compressed print	[v] ESC SHIFT Q [v]
to stop compressed	[v] ESC SHIFT N [v]
to print Large	[v] [n] [v]
to stop Large print	[v] [o] [v]
to **bold print**	[v] ESC ! [v]
to stop bold print	[v] ESC " [v]

That's all there is to it. For other printers, I would imagine you would have to substitute the appropriate printer codes.

As an added plus, in the process of this investigation I found out about the [G]lossary command and how to make my own TAB settings the default settings! As some sage once said, "If all else fails, read the directions."

I feel so good about this, I just may tackle VisiCalc - - again!

**This article was submitted in "hard copy" form and was retyped using a different text processor and printer. Imagine the fun!*

To clarify, GlossaryOne is a set of AppleWriter][Glossary files designed to access the special functions of Epson printers from AW][files

-ed.

CALL WITH PARAMETERS

-Reed Harris.

Use this program if you will. It uses the LAM technique to put a machine language subroutine into memory. And it works to do a fast "monitor memory move."

Line 88 clears the memory-move data sites.

Line 100 puts you in the monitar to test the move.

```
*** CALL WITH PARAMETERS ***
 10  REM  *** CALL WITH PARAMETERS ***

 12  REM  ** JULY 27 '83 **

 14  REM  ** SEE P38-39 IN 'THE GUIDE TO
     WHAT'S WHERE **

 16  REM  ** (C) 1983 BY REED HARRIS **

 18  HOME : VTAB 2

 20  PRINT  "THIS PART (LINES 20-100) SET
     UP  A CALL": PRINT : PRINT "WITH START
     &  END OF MEMORY-BLOCK TO MOVE": PRINT
     "AND  THE DESTINATION ADDRESS.": PRINT
     :  PRINT  "IN YOUR BASIC PROG. USE THE
     FORM":  HTAB 43: PRINT "XXX CALL 24576
     ,ST,EN,DE"

 30  PRINT : PRINT "LINE #70 HAS THE CALL
     ADDRESS.  CHANGE":  PRINT  : PRINT "IT
     TO  YOUR  IDEAL  LOCATION.  THEN  'RUN
     70'"

 40  LIST 70: STOP

 70  A$  =  "6000":  REM * HEX ADDRESS FOR
     CALL SUBROUTINE

 72  A$  =  A$ + ":20 BE DE 20 67 DD 20 52
     E7  A5  50  85 3C A5 51 85 3D 20 BE DE
     20  67  DD  20 52 E7 A5 50 85 3E A5 51
     85  3F  20  BE DE 20 67 DD 20 52 E7 A5
     50  85  42  A5 51 85 43 D8 A0 00 20 2C
     FE 60 00": REM  * MACH LANG PROG.

 78  A$  =  A$ + " N D823G": REM  RETURNS
     TO APPLESOFT

 80  FOR  I  = 1 TO  LEN (A$): POKE 511 +
     I,  ASC ( MID$ (A$,I,1)) + 128: NEXT :
     POKE 72,0: CALL  - 144

 82  END

 88  POKE  60,0:  POKE  61,0:  POKE 62,0:
     POKE 63,0: POKE 66,0: POKE 67,0

 90  INPUT "START : ";ST

 92  INPUT "END   : ";EN

 94  INPUT "DEST  : ";DE

 96  PRINT  "  START="ST"  END="EN"
     DEST="DE: CALL 24576,ST,EN,DE

 98  PRINT  PEEK  (60)"  "  PEEK (61)" "
     PEEK  (62)"  " PEEK (63)" " PEEK (66)"
     " PEEK (67)

100  CALL  -  151:REM DROP INTO MONITOR
     TO VERIFY THE MOVE
```

Epson Salts

Dumping It

(C) 1983 by Michael Fischer

Part 3 of a Series

When we left off last month we had learned a little assembly language, enough to set our own program between the normal input handling routines and the keyboard, ready to trap any Control-P's and activate the screen dump program. Of course, we conveniently left out the screen dump program itself. After all we needed something to fill up this month's *Cider Press*.

Those of you with a good memory or a good filing place for old *Cider Presses* know from part 1 of these articles that our program for a screen dump written in BASIC (slightly modified from part 1) is:

```
10  PR#1
20  S=1024 : E=1920 : GOSUB 70
30  S=1064 : E=1960 : GOSUB 70
40  S=1104 : E=2000 : GOSUB 70
50  PR#0
60  END
70  FOR I = S TO E STEP 128 : GOSUB
    80 : NEXT I : RETURN
80  FOR J = I TO I + 39
90  PRINT CHR$(PEEK(J));
100 NEXT J
110 PRINT
120 RETURN
```

Let's translate this into assembly language.

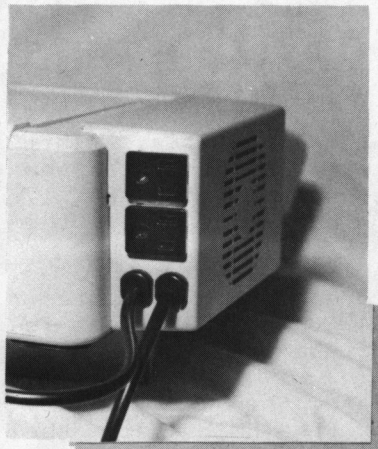
We'll begin with the easy stuff, lines 80 to 120. This requires learning several new commands and a new addressing mode.

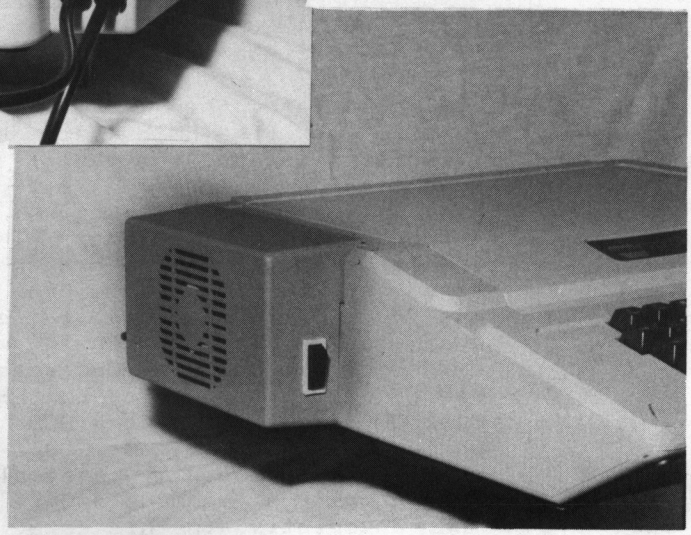
Modus Addressandi

"Wait a minute!" you all shout at once. "What's an addressing mode?"

As you recall from last time (quick scurrying to the back issue file) whenever we tell the computer to load the accumulator with a number, or store the number in the accumulator, we need to tell the computer where to get the number or where to place it.

Last time when we loaded a number into the accumulator we simply included the number in the program itself. This is the simplest addressing mode and is known as the immediate mode. Its symbol is the number sign (#).





Le Fan

pour la Pomme

-surge protection -matching colour
-illuminated switch -convenience outlets

$37.00 + tax

Orders may be placed and picked up at the General Meetings or by mail.

order from:
The San Francisco Apple Core, Inc.

1515 Sloat Blvd. Suite 2
San Francisco, CA

mail orders add $3.00 for shipping and handling.

When the number in the accumulator is stored, it must be stored in a location -- obviously the immediate mode cannot be used. Last time we used the direct mode which is simply the address of the location preceeded by a dollar sign ($) to indicate that it is a hexadecimal number.

Page Zero

Actually there are two forms of the direct addressing mode -- regular and page zero. Regular uses any address other than one on page zero. And page zero is any address on page zero. But what is page zero?

Take any convenient book and open it to page zero. That's funny -- there's no page zero in my book either. But there is one in the Apple memory.

Apple's memory contains pages of memory, each 256 ($100) bytes long. The first page of memory, located at addresses 0 to 255 ($00 to $FF) is called, quite logically, page zero. For reasons unnecessary to go into here, page zero locations are considered special.

Another addressing mode is the indirect indexed method. This method uses the Y register to determine the address from which data is to be obtained or to which it is to be stored.

Why Register?

I bet you're wondering what the Y register is. We already are familiar with the accumulator as an area in the 6502 processor which can hold any one-byte number. In addition there are two special purpose areas, known as the X and Y registers, which can also hold a one-byte number. These registers can be used just like the accumulator (except addition and subtraction can only be done to the accumulator); they can also serve special purposes.

The indirect indexed method of obtaining an address uses an address stored on page zero of memory as a base, and then adds the number in the Y register to get the true address. *E.g.*, if the page zero location contained the address 2514 and the Y register contained the value 25, then the true address would be 2514 plus 25 or 2539. By incrementing the Y register a whole section of memory can be used, one location at a time.

Two Bytes

The first step in translating lines 80 to 120 into assembly language is placing the value of I into a page zero memory location. The first value of I is 1024 ($400) which is large than one byte. (As you may recall, a single byte can contain any number up to 255 ($FF). A larger number, up to 65535 ($FFFF) must be stored in two bytes. This is done by dividing the number by 256 and placing the integer value of the result in one of the bytes (called the high byte) and putting the remainder (or the number modulo 256) into the other byte (called the low byte). This number is stored with the *low* byte first.

To store 1024, then, we first divide it by 256. This gives a result of 4 with a remainder of 0. The low byte is 0 and the high byte is 4 and it is stored in memory as 00 04.

Any available two-byte page zero memory location can be used to store the value of I. The particular location used in this program is called IVAL and is located at 237 to 238 ($ED-$EE). By inserting the assembler command

```
IVAL EQU $ED
```

in our program we instruct the assembler to use memory location $ED whenever it sees IVAL in our program.

Saying It Indirectly

The proper way of showing the indirect indexed mode of addressing is

```
LDA (IVAL), Y
```

This tells the microprocessor to load the accumulator with whatever number is located at the address obtained from adding the number in the Y register to the two-byte number in IVAL and IVAL + 1.

Four new commands are needed for this section of our program. They are:

LDY - for LoaD the Y register. This is just like LDA except the number is placed in the Y register instead of the accumulator.

INY - for INcrement the Y register. This adds 1 to whatever value is in the Y register.

CPY - for ComPare the Y register. This is just like CMP except the Y register is used instead of the accumulator.

BLT - for Branch if Less Than. This is like BNE except the branch is taken if the number is the Y register (or accumulator) is less than the number to which it was compared. (Note: If your assembler does not recognize the BLT op-code, substitute the BCS or Branch if Carry Set code. BLT is used here because my assembler recognizes that command and it is more descriptive in this case.)

And Away we Go

LDY #0
The Y register will hold our loop value to be added to the "I" value.

LINELOOP LDA (IVAL), Y
Get the character at "J" (IVAL + Y) into the accumulator.

JSR COUT
Go to the subroutine that prints the ASCII character whose value is in the accumulator. (This subroutine is described later in this article.)

INY
Increase the Y register by one which will increase the "J" value. This is like the NEXT J command.

CPY #40
If Y is 40 or more, then "J" is more than "I" + 39 and we need to exit the loop.

BLT LINELOOP
If Y is less than 40 continue the loop.

JSR CROUT
If we are done with the line we need to tell the printer to print a carriage return and a line feed. The CROUT subroutine (described later) does this. This is equivalent to line 110 in the BASIC program.

RTS
And we are done. (Line 120 in the BASIC program.)

Two Subroutines

The COUT (for Character OUTput) subroutine takes whatever value is in the accumulator and places it in a special location on the Epson interface card. The interface card then sends that character to the printer and when it is done sets the value of another location to a value of less than 128 ($80).

COUT STA $C090
This location, which is 41104 in decimal, is the printer output location on the Epson interface card.

GOTIT LDA $C1C1
This location, which is 41377 in decimal, contains a value greater than 127 until the character in location $C090 is accepted by the printer. It then changes value to less than 128.

BMI GOTIT
Numbers from 128 to 255 ($80 to $FF) are considered negative numbers while numbers from 0 to 127 ($00 to $7F) are considered positive. BMI stands for Branch if MInus. This line loops back to GOTIT for as long as the number in $C1C1 is negative.

RTS
And back to the calling routine.

Carriage Return

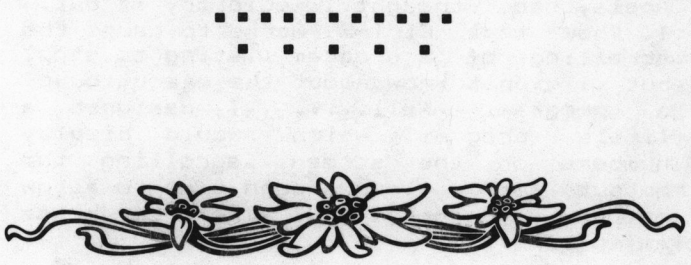
The CROUT (for Carriage Return OUTput) subroutines sends a carriage return and line feed to the printer using the COUT subroutine.

CROUT LDA #$0D
The value 13 ($0D) is the ASCII value of a carriage return. It causes the print head of the printer to return to the left margin.

JSR COUT
LDA #$0A
The value 10 ($0A) is the ASCII value for line feed. It causes the printer to move the paper up one line.

JMP COUT
And we return, via the RTS of COUT, to the calling routine.

(Next time we will continue to translate our BASIC program to assembly language.)



AUTHORS WANTED

Share your experiences, Apple programming hints, pascal tips, software reviews. Send your material to the Cider Press Editor at the Clubhouse.....

B*A*S*I*C

Copyright (c) by Joyce Conklin, 1983

The first item for this month is a QUESTION!! In reading over some material relating to the IBM, it seems there is a Ctrl-Numlock combination which will cause a program to pause in execution by the operator pressing the keys; it does not have to be built into the program. Then it says that you can resume operation by pressing any key.

That sounded a bit like Ctrl-S on the Apple, so I thought I would try it out. I knew that Ctrl-S works to cause the scrolling of a program listing to stop, but I didn't know about the execution of a program. Well..... I designed a little program which would display numbers on the screen, scrolling the output for a long enough time to allow me to try the Ctrl-S. The program was as follows:

```
10 HOME:FOR L = 1 TO 1000:X = INT(RND
   (1)*10):PRINT X, "  ",:NEXT L:END
```

Notice that the commas will make the output appear in 3 columns, and for this program the Ctrl-S works to stop the scrolling. Notice also that the 3 columns are double-spaced. What is actually happening, using the above listing is that a number is printed in column 0, 2 spaces in column 17, a number in column 33, 2 spaces in column 0, etc.

If you change the listing to delete the 2 spaces between the quotes (just have PRINT X,:) you will still have 3 columns and Ctrl-S will work, but the numbers will not be double-spaced. However...!! If you change the listing to show ...

```
*10):PRINT X;"  ";:NEXT ...,
```

strange things happen. Sure enough, the numbers print with only 2 spaces between, but now the Ctrl-S DOESN'T work. Try it all you want and the numbers just keep on scrolling until you press Ctrl-C, which causes a BREAK IN 10 message to appear and the program to stop entirely. Does anyone have an explanation for this?

. . . . .

The first approach to solving the July problem about data storage is to determine the formulas needed. The gap between records is 30 mm or 3.0 cm or .030 m. It is a good idea to have all your constants of the same type of measurement (linear distance or time, or volume, or whatever) in the same units (e.g. all metres or all millimetres, etc.) You can fit 750 bytes in one centimetre (DEnsity = 750.) REcord = 1500 bytes, so it will take 2 cm or 0.02 m to hold one record.

Add in the gap distance, and the total LeNgth of one record is 5 cm (2 cm + 30 mm). It seems that this particular tape format requires more gap space than recording space! When you realize that the tape has to stop and start between each record, this explains the need for a long non-recorded distance.LEngth of tape (1500 m) divided by LeNgth per record = number of records on tape (1500/.05 = 30000 records.)

Now you know the answer - but you could have done that with a calculator! The purpose of a computer program is to have a way to solve similar problems which do not have exactly the same numerical input.

```
10 REM DE = NUMBER OF BYTES/CM

20 REM G  = DISTANCE OF GAP IN MM

30 REM RL = RECORD LENGTH (BYTES)

40 REM TL = TOTAL LENGTH OF TAPE(METRES)

50 REM RD = LENGTH OF RECORD (CM)

60 READ DE,G,RL,TL

70 DATA 750,30,1500,1500

80 G = G/10: REM GAP DIS. (CM)

90 RD = RL/DE + G

100 RD = RD*100: REM REC.DIS.(METRES)

110 HOME : VTAB 5 :PRINT"NUMBER OF
    RECORDS PER TAPE =";TL/RD

120 PRINT "OR ";(TL/RD)*1500/1000;
    " KILOBYTES"

130 NR = TL/RD: REM NUMBER OF
   RECORDS/TAPE
```

```
140 READ RS, VC: REM RS= READING SPEED
    VC = TIME FOR VELOCITY CHANGE

150 DATA 3, .001

160 RT = RD/RS: REM READ TIME PER RECORD

170 RT = RT + 2*VC: REM ADD IN START AND
    STOP TIMES

180 TI = NR*RT: TI = TOTAL READ TIME

190 VTAB 8: PRINT "TOTAL READ TIME = ";
    TI

200 READ VL: REM SIZE OF ENCYCLO.(BYTES)

210 DATA 150000000

220 TT = VL/(TL/RD*1.5):REM TOTAL TAPES

230 VTAB 10: PRINT" TOTAL TAPES FOR THE
    ENCYCLOPEDIA = ";TT

240 VTAB 12: PRINT "THE ENCYCLOPEDIA
   WILL NEED ";VL/143000;" APPLE FLOPPY
   DISKS"

250 END
```

This way, you can change the DATA items to whatever values you wish, although you must be careful to use the dimensions or units specified; otherwise the calculations in the program will result in at least an error of magnitude (decimal point error) if not a completely wrong answer.

. . . . .

To keep you busy for the rest of the summer, here is a problem which I gave as one of the options for the semester final program in Computer 1 this past term. Only a very few attempted it, but those who did, did well. You should rememeber the following statements to do this problem:

DIM, READ, DATA, as well as the usual ones. You may also have to refer to the column on sorting to refresh yourself about the bubble sort.

12. (12 points) Consider an N-question multiple-choice exam, where only one answer is correct for each question.

Write a flowchart and program to grade the exam and print the student's name, the number right, the number wrong, and the final grade. The final grade is the percent number right. The data consist of the following:

1. One line for N
2. One line containing N integers representing the N correct answers (the "answer key")
3. One line indicating the number of students (S) in the class who took the exam
4. S lines, one for each student, containing the student's name and N answers.

Use two one-dimensional arrays to store the answer key and a student's answers. Test your program with the following test data.

```
20   (number of questions)

1,3,5,4,4,1,1,2,3,5,5,1,2,3,4,4,5,3,2,1
    (answer key)

5    (number of students)

"Petrocelli",1,3,4,4,4,1,1,2,3,3,5,1,2,
          3,4,4,5,1,1,1

"Baker",1,3,5,4,3,2,1,2,3,4,3,2,2,3,3,3,
        5,1,2,1

"Valentino",1,3,5,4,4,1,1,2,3,5,5,1,2,3,
          4,4,4,3,2,1

"Simpson",1,2,5,4,4,2,2,3,3,1,2,1,3,3,3,
          5,5,3,1,1

"Carter",2,3,5,4,4,1,1,3,3,5,4,1,5,4,4,4,

          5,2,1,3
```

12a. (6 points) In addition to #12, print the name and grade of the student with the highest grade.

12b. (9 points) Do problem #12, but modify the output such that the names of the students appear in alphabetical order. (Hint: The bubble sort routine can be used for alphabetical as well as numerical data.)

12c. (6 points) In addition to #12, print a frequency distribution of the final grades for the exam, as follows:

```
90 or above      xx
80 to 89         xx
70 to 79         xx
60 to 69         xx
50 to 59         xx
Below 50         xx
-------------------
                 xx

Average Grade for Exam = xx.x
```

(This problem is from "APPLIED BASIC PROGRAMMING", by Roy Ageloff and Richard Mojena, publ. Wadsworth Publ. Co. 1980)

. . . . .

This should keep you working at your Winesap between your excursions to the beach or the trout stream or wherever you spend your summer vacation. Have fun!

Pascalia and Fugue

ADDRESSING ENVELOPES

by

Max J. Nareff

In a recent issue a simple method of using the computer as a typewriter was publicized, directly from pg. 38 of the Apple Pascal Operating System Manual. Here now, is a more detailed method of using the computer to address envelopes. It lets the computer and printer to do most of the work of positioning the printer head. The program provides for both the return and mailing address.

```
(*Used for addressing envelopes*)
PROGRAM ENVELOPES;
VAR
  I,
  LINES,      (* # of Lines in Address *)
  FIRSTLINE,  (* On the Envelope *)
  INDENT,     (* Left Margin *)
  VERT        (* ON THE SCREEN *)
                :INTEGER;
  F : INTERACTIVE;
  STR : STRING;
  CH : CHAR;

PROCEDURE GETASTRING
   (VAR S:STRING;     (* String Input *)
    X,Y,              (* GotoXY Coord *)
    MAXLENGTH:INTEGER);
        (* Desired Length *)

VAR
  I,J : INTEGER;
  CORRECT : BOOLEAN;
  BELL : CHAR;
BEGIN
  BELL := CHR(7);
  CORRECT := FALSE;
  REPEAT
     GOTOXY (X,Y);
     FOR I := 1 TO MAXLENGTH DO
        WRITE ('-');(* String Template *)
     GOTOXY (X,Y);
     READLN (S);
     IF (LENGTH (S)>MAXLENGTH) THEN
        BEGIN
           GOTOXY (0,23);
           WRITELN
   ('String too long! Please Repeat':35)
        END
     ELSE
        CORRECT := TRUE
  UNTIL CORRECT
END; (* GETASTRING *)


(* Adapted from "Pascal Programming for
the Apple" *)  (* By T.G. Lewis, 1981,
Reston Pub. Co. *)
FUNCTION TAB (X:INTEGER):CHAR;
VAR
  I : INTEGER
BEGIN
  FOR I := 1 TO X DO
     WRITE (F, ' ');
  TAB := CHR (15); (* Does nothing *)
END;   (* TAB *)

BEGIN  (* Main *)
  PAGE (OUTPUT);
  WRITELN ('Direct Envelope Addresser.');
  REPEAT
     REPEAT
        WRITELN; WRITE
    ('How many lines in address? 1..5  ');
        READLN (LINES)
     UNTIL LINES IN [1..5];
     REPEAT
        WRITE ('Starting line? 1..10 ');
              (* On the envelope *)
        READLN (FIRSTLINE)
     UNTIL FIRSTLINE IN [1..10];
     REPEAT
        WRITE ('Left Margin? 0..40');
        READLN (INDENT)
     UNTIL INDENT IN [0..40]; WRITELN
     REWRITE (F, 'PRINTER:');
     FOR I := 1 TO FIRSTLINE DO
      (* Positions printer on envelope *)
        WRITELN (F);
     VERT := 10;
    (* Vertical pos'n for
       screen appearance only *)
     FOR I := 1 TO LINES DO
        BEGIN
           GETASTRING (STR, 23, VERT, 35);
           VERT := VERT + 1;
           WRITELN (F, TAB (INDENT), STR)
        END;
     CLOSE (F);
     PAGE (OUTPUT);
     WRITELN; WRITE ('Another Entry on same
              envelope (i.e. address)?');
     READ (KEYBOARD, CH);
  UNTIL CH IN ['N', 'n']
END.
```

ERRATA (tat-tat)

The Pascal article entitled:
Pascal Etiquette, A Graceful Way Out
THE key line was left out.
Following the comment line:

```
(*This  is  where the abort signal comes
in*)
```

the following line should be inserted:

```
IF KEYPRESS THEN
```

The San Francisco Apple Core, Inc.

invites you
to its third

Semi-Annual Swap Meet

Building C - Fort Mason
San Francisco, CA

September 10

Room 100
10:00 A.M. to 2:00 P.M.

Silent Auction - (ends 1:00 P.M.)

Dealer Auction Starts 1:15 P.M.

Consignment Table

Booth Space Available to Individual Members $25.00
Booth Space Available to Dealers - - - - - $75.00

Members - - - Free
Non-Members - $1.00

For Sales Information:

Contact Don Scellato (415) 672-2051 (*evenings*)

Aztec C

a review by David B. Anderson

Product Review: Aztec C by Manx Software

First I'll describe what the package is, then how it performs, then give my evaluation.

C is the programming language used to develope the Unix operating system. This package contains not only a full C compiler (two of them!) as defined in the book "The C Programming Language" by Kernighan and Richie but a comprehensive set of subroutines and a rather Unix-like replacement for the Apple DOS command processor.

One compiler generates interpreted code for an "interpreter assembler" and an interpreter is provided in a library. The other compiler generates 6502 assembler and a 6502 assembler generates linkable object. Which brings up the Linker. The linker can take output from either assembler - including any routines written directly in assembler - and create an executable binary file. The linker selectively links from libraries of subroutines if you wish.

The following example shows the commands needed to
compile and link and execute the program used as the timing example below:

```
c65 prime.c

(switch discs to the linker disc)

ln prime.rel sh65.lib,d1

prime
```

Unfortunately the compiler and linker will not fit on one disc!

The editor is called "ved" and is a small but adequate relative of the "vi" editor written at UC Berkely. The screen driver provided makes it possible to handle both upper and lower case characters and to generate the special characters needed by C on an unmodified Apple II. Escape does shift lock/unlock, for example. And upper case characters are displayed in reverse video. Lines longer than 40 columns wrap to the next line and aren't any problem. A Videx 80 column board is reputed to work and a driver is provided for it. Information is provided - including driver source - to make it possible to write one's own device driver. Astonishingly, the editor is provided in source too. I have not tested the editor's limits but the documentation says a file of 25000 characters can be editted.

The documentation is critical, of course. As provided it is useful and useable. It is NOT tutorial. As an experienced Unix user I have had no problems. But read on.

INTERNATIONAL APPLE CORE ™

APPLE ORCHARD **SUBSCRIPTIONS**

For IAC Club Members Only

908 GEORGE STREET, SANTA CLARA, CALIFORNIA 95050

☐ 9 issues for only $19.50
($29.25 at newsstands)
$17.00 for members of IAC clubs only

☐ Foreign: 9 issues
$36.00 Surface
$29.00 for members of IAC clubs only

☐ Payment enclosed
(US funds only)

Start my subscription with issue # _____

Name _____

Address _____

City _____ State _____ Zip _____

☐ Name of Club _____ Signature of club officer _____ Valid only with signature

InterBank # ☐☐☐☐ 4 digit number above your name on MasterCard™ ☐ VISA

Charge Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Credit Card Signature _____ Card Expires

The documentation covers not only operational details but also the large set of subroutines. All the calls a C programmer is used to using are there. Plus many that may not be used much. These subroutines are provided in source too!

The documentation is provided in a classy 3-ring binder and is available for purchase ($30) by those who wish to look before plunking down their $199. I bought the documentation seperately first and then bought the package.

The system uses DOS as its underlying mechanism. This means all the files are regular DOS files. The normal system discs can be used for copying as desired. The compiler etc discs are not copy protected so you can make those backups before starting work. Moreover one can write C programs and access Basic source code or anything else that is on a normal DOS file.

The system is supplied on 6 discs. I have a 2 drive system and leave my source in one drive and switch back and forth between the a compiler/editor disc and the linker disc. A 3 drive system would be easier to work on but I haven't minded the disc switching.

Important point number 137. 64K memory is required! I have an Apple Language Card. Any of the cards extending memory to 64K are supposed to work.

One can't talk C without talking performance. I have chosen to code the Sieve of Eratosthenes program as in the September 1981 Byte Magazine article by Jim Gilbreath (Page 180) called "A High Level Language Benchmark".

The 6502 code generator compiler compiled and assembled in 102 seconds. Linking took 98 seconds more. Load time took about 7 seconds. Execution (using the full 10 iterations) took 34 seconds. Compare that with the reported Integer Basic compiler time of 0 and run time of 2320 seconds. And UCSD Pascal compile of 43 seconds and run of 516 seconds. Z80 times run as low as 14 seconds execution (Digital Research PL/1-80) to as high as 5115 seconds (Microsoft COBOL version 2.2) execution. Enough numbers. To my surprise the C interpreter took about the same run time as the 6502 native code generator for the Sieve (34 seconds).
The executable C code was about 6k bytes for the 6502 code. The interpreter generated under 3k bytes. Both sizes are final run-time code sizes reported by the linker with subroutines from the libraries included.

I think this is a fine package and well worth the money. It helps that I have found no bugs so far. I do have reservations, however.

If you are familiar with Unix and C and like that environment you'll be delighted with Manx C as I am. If you don't know the environment you will find it hard to learn. Probably impossible without expert help from someone who does know the Unix/C environment. The documentation is great if you already know what to do and approximately how to do it. The problem is not unique to Manx Software. I have simply found that everyone new to the environment (even on a $250,000 PDP11/70!) has trouble getting started. Unix manuals seem to be useless as learning tools and the Manx manuals are only somewhat better than regular Unix manuals.

Prices: Manual only $30.

Manual plus software (Licensing agreement allows reproductions of software you write but NOT giving away the compiler!) on 6 5 1/4 inch discs: $199.

Available from:

Manx Software Systems
P O Box 55
Shrewsbury N J 07701

UNIX is a trademark of Bell Labs
This product review written by David B. Anderson. July 6, 1983

Wanted

WANTED: Apple II System or Parts
Are you a local business or individual who wants to trade in your old Apple on a new IIe? If so, consider donating it to the Science Department at Redwood High School, Larkspur. Our students would be very grateful and Uncle Sam would reward you by a suitable reduction in income tax. Your friendly Apple dealer would probably cooperate in this effort to help a local school. I'd like to talk to you even if you are only interested in making a donation, but not a trade in. Contact: John Girton, 15 Del Norte, San Anselmo, CA 94960. Tel: (415) 457-2647.

MARKETING DEPARTMENT NOTES
by
Don Scellato

I would like to thank each member who worked at the Northwest Computer Swap Meet at the Oakland Convention Center on July 23. This Swap Meet exposed the Club to a number of prospective members and we managed to sell some of our product at the same time. Ben Lai deserves a special Thank You for the terrific signs he made for use at the Swap Meet and at similar affairs.

Look for a sign up sheet for the Third Semi Annual San Francisco Apple Core Swap Meet which will be held in Building C, Fort Mason, from 10 A.M. till 2 P.M. on Saturday, September 10. This Swap Meet will take the place of our regular September Meeting. We can use some help at the Club Sales Table, so anyone who is interested in helping should sign up at the August Meeting or should call me. Anyone interested in running his own booth to sell new, used, or overstock items should contact me immediately as booth space will be given out on a first signed up, first served basis. Booths for members will cost $25. Booths for dealers will cost $50. We will run a consignment table right next to the Sales Table. If you have some hardware or software you no longer need, you can sell it there. Just bring it in by 9:30 A.M. so we can log it in and set it up for sale. Non members will be charged $1 entry fee, but members can attend free.

There will be another Swap Meet in San Mateo in October. If anyone is interested in assisting us with a Sales Table at that event, please call me.

You might be curious about what is included on the Machine Language Utility Disk that we started selling in June. This disk contains a tutorial on Machine Language, a Mini Assembler, and several utility programs for Machine Language programming. It's a relatively easy way to get into that particular aspect of programming.

Once again, I direct your attention to the order form for commercial software which appeared in last month's Cider Press. If you are interested in saving money on any of that software or any other software or hardware, contact me at the August Meeting

Contest Notes

by
Don Scellato

By the time you read this article, the Programming Contest will be over and the Contest Committee will be trying to determine who deserves each of the fine prizes donated to the Club by the Microcomputer Industry. If you've entered the contest, come to the August meeting to find out whether you've won a prize. If you haven't entered, come to the meeting anyway to share the thanks of all of the membership with those members who have taken the time to write a program or game which may be shared with the rest of the Club's members. Next month's issue of the Cider Press will feature a brief write up about all of the entries and will list the prize winners. Unfortunately, we're going to press too early to include that material in this issue.

In the meantime, the Board of Directors wishes to take the time to thank all of you who entered the contest for the work you put in on your entry.

Adopt a School

As mentioned in the July Cider Press we have been invited to participate in Apple's Adopt A School Program. The purpose of this program is to provide the schools with a volunteer resource base to assist them with their initial orientation and set-up.

I have been receiving calls from interested volunteers and will need more help if the program gets off the ground. At this time we haven't received the list of schools from Apple Computer so it is uncertain what the club's involvement will be. If you are interested in participating please give Sami Capron a call for more information (415-388-8631).

Interest Groups

Telecommunications

Communications interest group

July meeting
S.F.A.C. Club house
27 July 1983

****
****

This month we had a smaller group than usual, possibly due to the fact that a lot of members are on vacation. As a follow up to the discussion at the May general meeting, we discussed splitting up the group into regional groups (North bay,South bay). The general feeling was that the group is not yet well enough established to survive a split up.

It was decided that the location of the meetings will rotate between three locations. (North bay, South bay & East bay). In this way we hope to make the group more accesible to all members in the bay area. We will continue to meet at the club house until we have located suitable meeting locations in the South & East bay.

Some of the group members have had some trouble using the Apple IIe with the 80 col. board at 1200 baud. The above combination tends to lose characters here and there. We would like to hear from anyone else who has encountered this problem & its solution (if they found one). Please drop a note to us at the club house or come to the next meeting.

What is E-COM ?

E-COM (Electronic Computer Originated Mail) is a new service introduced by the U.S. Post Office in 1982. Under the traditional system, mail (hard copy in an envelope) is introduced into the mail system at the senders end. E-COM allows the user to generate messages on a computer and transmit them to any one of 25 SPO's (Special Post Office) strategically located across the country where the letters are printed, placed in envelopes & introduced into the First Class Mail system.

Delivery is made in substantially shorter time. Messages are limited to two pages in lenght at a cost of $.26 for the first page & $.05 for the second page. There is an annual fee of $50.

There are two classes of access facilities - dial up & dedicated. Dedicated access is designed for those with a large volume of mail. The dial up access allows the user to connect to the SPO via a modem, the telephone & one of the common telecommunications carriers. Asynchronous communication will be possible at speeds of 300 and 1200 baud, full duplex.

We still need the names of some European commercial on-line Data Bases (such as The Source & Compuserve). Perhaps some of our European members could drop us a note?? (use the club house address)

Listed below are the dates of upcoming meetings. At present, meetings are held at 7.30 P.M. on the fourth Wednesday of the month at the club house; however, as indicated above, the location of the meeting will be subject to change in the future. The new locations will be published in this column as soon as they become available.

24 Aug- Bob Teicher of Dow Jones will demonstrate the Dow Jones news & information service.

28 Sept- A demonstration of a local area network is being arranged forthis date

++++++++

VISICALC INTEREST GROUP
by
Don Scellato

The next meeting of the San Francisco branch of the VisiCalc Interest Group will meet at the Clubhouse at 7:30 P.M. on Tuesday, August 9, 1983. We will be dealing with the use of the M&R SUP'R TERM 80 column card with The SPREADSHEET and will be happy to answer any questions any of you might have about VisiCalc or to demonstrate anything you might not understand about the program.

The East Bay VisiCalc Interest Group will meet at Breuner's Furniture Store in Pleasant Hill in the Meeting Room on the Second Floor from 7:00 P.M. till 9:00 P.M. on Thursday, August 25. We will try to bring in an Apple III computer to explore the features of The Advanced Version of VisiCalc. If we can't borrow an Apple III, we will do some work with Multiplan CP/M version.

INTEREST GROUPS
By
Sami Capron

The summer has been quiet for the Interest Groups. Hopefully we will get more interest generating in the fall. We still need volunteers to lead new groups. I can be reached at 388-8631 if you are interested.

There are a few changes to note regarding the interest groups. The Nu'ers group in Marin no longer meets regularly at 55 Main St. in Tiburon. I would like to take this opportunity to thank Reed Harris for his generousity in allowing us to use his office. If there is anyone who has space available for the Nu'ers let me know. Also for up to date information about the group contact me at 388-8631.

COMPUTER CAMPS
By
Sami Capron

We have all heard about the computer camps that have been popping up all over the place. I recently had an outstanding experience with such a camp that I thought I would share with the membership.

My work involves supervising young people that have had some trouble with the law. As I see it, my work also involves rechanneling some of their energy in a positive and constructive manner. Well, I got the brain storm to acquire a computer campership for one of my young people. I called every camp I could find in California and found only one camp willing to provide such a scholarship.

I would like to take this opportunity to applaud the DATA BASE COMPUTER CAMP in Calistoga for their outstanding effort with my young person. I was equally impressed with their facilities and the other camp activities that were offered. I would highly recommend this camp experience for any young person.

I thought members would like to know that Mr. and Mrs. Abad, the Camp Directors are also involved in adult seminars in the East Bay. I understand they also conduct periodic seminars in Hawaii. For more information contact Valerie Abad at 6454 Valley View Road, Oakland (415-836-1292 or 415-339-2961).

San Francisco APPLE Core
Statement of Cash Receipts and Disbursements

Year ending March 31, 1983

Recipts:		
Dues		$25216
Sales Table		23385
Advertising		2417
Mail Order & Other		6729
		$57747
Disbursements:		
Cider Press		19771
Membership		2823
Marketing		
Merchandise	14129	
Rent	2752	
Tax	1843	
Promotion & Other	2227	20951
Office		
Accounting & Personnel	7700	
Rent	3300	
Postage & Phone	1505	
Ballots	911	
Legal	494	
Other	2531	16441
Total Disbursements		59986
Net Cash Flow		-2239

August 1983

Monday	Tuesday	Wednesday	Thursday
	2 East Bay 7:30 Forth 7:30		4 Green Apples 8:00
8 Pascal 7:30	9 CP/M 7:30 VisiCalc 7:30	10 Board of Directors 7:00 Marin Graphics 7:30	
	16 Forth 7:30	17 Apple /// 7:30	
22 Pascal 7:30	23 CP/M 7:30 Epson 7:30	24 Telecomm'ns 7:30 31 Board of Directors 7:30	25 Green Apples 8:00 VisC-East Bay 7:00

September 1983

Monday	Tuesday	Wednesday	Thursday
			1 Green Apples 8:00
	6 East Bay 7:30 Forth 7:30		
12 Pascal 7:30	13 CP/M 7:30 VisiCalc 7:30	14 Marin Graphics 7:30	
	20 Forth 7:30	21 Apple /// 7:30	22 Green Apples 8:00 VisC-East Bay 7:00
26 Pascal 7:30	27 CP/M 7:30 Epson 7:30	28 Telecomm'ns 7:30	29 Green Apples 8:00

INTEREST GROUP LEADERS

Apple ///	Don Norris	673-7635 (day)
CP/M	Jim Ayers	383-0473(day) 383-5759(eves)
Epson	Michael Fischer	557-1250 (day) 933-4435(eve)
Forth	Dudley Ackerman	626-6295
Green Apples (BASIC)	Gary Owens	521-4002
Marin Graphics	Reed Harris	924-7221
	Bob Sachs	332-3331
Nu'ers	Sami Capron	388-8631
Pascal	Mike Schnapp	386-2238
Product Review	"Woody" Liswood	585-6375
Telecommunications	Frank Fennelly	(408)984-7321
VisiCalc	Don Scellato	672-2051(eve)

APPLE CORE OFFERINGS

NOTES

SFAC 1983 DISK NUMBER 1 contains only three programs due to the nature of one of the programs - a hi-res graphics game. This game is called "The Incredible Shrinking Man" which was contributed by the Chiang Brothers. Also included on the disk is a High Score Tabulator and hi-res graphics display program called Gallery.

LIBRARY #1 contains over 340 programs. It is packaged in diskette holders and is bound in a SF Apple Core binder with over 30 pages of documentation. This documentation alone is available for three dollars.

LIBRARY #2 continues the tradition started with Library #1. It contains over 250 programs from the 1980 DOMs.

1981 YEARBOOK contains 12 disks containing all DOM programs from 1981. There is a heavy emphasis on games and business applications.

1983 PROGRAMMING UTILITIES contains a collection of utilities not collected on a single disk before. These utilities come from a variety ofsources such as IAC disks, 1981 DOMs, and 1982 DOMs.

1983 GAME SET consists of two disks containing games not collected in a single set before. It contains a variety of hi-res, lo-res, and text games.

1983 EDUCATION APPLICATIONS is a set of two disks containing interactive learning programs, a grade maintenance program, and GRAMMAR QUEST an interactive learning game.

1983 HOME APPLICATIONS consists of two disks of programs useful around the home. One disk was from the 1982 Home Pak. The second disk is a collection of programs gathered from IAC disks and other recent contributions.

SFAC 1983 DISK 2 - this disk contains a collection of games, home applications, education applications, and utilities.

IACBASE - this is a DataBase Manager provided by the International Apple Core. It is written in run time PASCAL and provides the user with an inexpensive DataBase Manager.

MACHINE LANGUAGE TUTORIAL/UTILITIES - this disk teaches the principles of Machine Language Programming and provides the user with some convenient programming utilities.

PRODUCT	MTG	MAIL
Library #1	98.00	101.50
Library #2	98.00	101.50
1981 Yearbook	98.00	101.50
DOMs		
Mar 82	5.00	7.00
Apr 82	5.00	7.00
May 82	5.00	7.00
COLLECTIONS		
Basic Tutor Pak(3)	20.00	22.00
Forth Pak (3)	20.00	22.00
Game Set (2)	15.00	17.00
Home Applications(2)	15.00	17.00
Education Applic.(2)	15.00	17.00
1983 Prog Util.(1)	10.00	12.00
SFAC Visicalc Temp(1)	10.00	12.00
1983 Visicalc Util(1)	18.00	22.00

All foreign orders: Add 25% to above prices.

SFAC MAIL ORDER BLANK

NAME.....

ADDRESS.....

CITY.....STATE.....

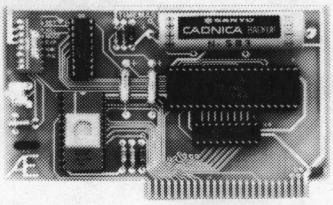
ZIP.....

Apple Peripherals Are All We Make

That's Why We're So Good At It!

NEW

The TIMEMASTER
Finally, a clock that does it ALL!



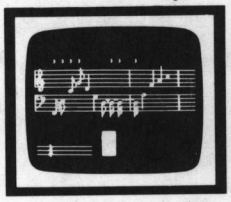
- Designed in 1983 using I.C. technologies that simply did not exist when most other Apple clocks were designed.
- Just plug it in and your programs can read the year, month, date, day, and time — down to 1 millisecond!
- Powerful 2K ROM driver — No clock could be easier to use.
- Full emulation of most other clocks, including Mountain Hardware's Appleclock (but you'll like the TIMEMASTER mode better).
- Compatible with all of Apple's languages, CP/M and PASCAL software on disk.
- Eight software controlled interrupts so you can execute two programs at the same time. (Many examples are included)
- On board timer lets you time any interval up to 48 days long down to the nearest millisecond.

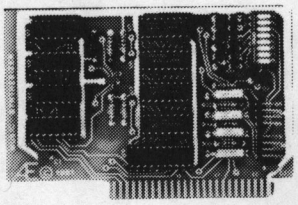
The TIMEMASTER includes a disk with some really fantastic time oriented programs (over 25) plus a DOS dater so it will automatically add the date when disk files are created or modified. This disk is over a $200.00 value alone — we give the software others sell. All software packages for business, data base management and communications are made to read the TIMEMASTER.

If you want the most powerful and the easiest to use clock for your Apple, you want a TIMEMASTER.

PRICE $129.00

Super Music Synthesizer



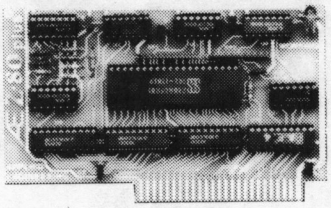


- Complete 16 voice music synthesizer on one card. Just plug it into your Apple, connect the audio cable (supplied) to your stereo, boot the disk supplied and you are ready to input and play songs.
- It's easy to program music with our compose software. You will start right away at inputting your favorite songs. The Hi-Res screen shows what you have entered in standard sheet music format.
- Now with new improved software for the easiest and fastest music input system available anywhere.
- We give you lots of software. In addition to Compose and Play programs, the disk is filled with over 30 songs ready to play.
- Easy to program in Basic to generate complex sound effects. Now your games can have explosions, phaser zaps, train whistles, death cries. You name it, this card can do it.
- Four white noise generators which are great for sound effects.
- Plays music in true stereo as well as true discrete quadraphonic.
- Full control of attack, volume, decay, sustain and release.
- Will play songs written for ALF synthesizer (ALF software will not take advantage of all the features of this board. Their software sounds the same in our synthesizer.)
- Automatic shutoff on power-up or if reset is pushed.
- Many many more features.

PRICE $159.00

NEW

Z-80 PLUS!



- TOTALLY compatible with ALL CP/M software.
- Executes the full Z-80 and 8080 instruction set.
- Fully compatible with microsoft disks (no pre-boot required).

- An on-card PROM eliminates many I.C.'s for a cooler, less power consuming board. (We use the Z-80A at a fast 3.58 MHZ)
- Does EVERYTHING the other Z-80 boards do, plus Z-80 interrupts.
- All new 1983 design incorporates the latest in I.C. technologies.
- Complete documentation included. (User must furnish software)

The Z-80 PLUS turns your Apple into a CP/M based computer. This means you can access the largest body of software in existence. Two computers in one and the advantages of both, all at an unbelievably low price.

COMING SOON: The Z-80 Plus for the Apple III

PRICE $139.00

Analog to Digital Converter

- 8 Channels
- 8 Bit Resolution
- On Board Memory
- Fast Conversion (.078 ms per channel)
- Eliminates the Need to Wait for A/D Conversion (just PEEK at data)
- A/D Process Totally Transparent to Apple (looks like memory)

The analog to digital conversion takes place on a continuous, channel sequencing basis. Data is automatically transferred to on board memory at the end of each conversion. No A/D converter could be easier to use.

Our A/D board comes standard with 0, 10V full scale inputs. These inputs can be changed by the user to 0, -10V, or -5V, +5V or other ranges as needed.

Information on temperature sensors is given in manual.

The user connector has +12 and -12 volts on it so you can power your sensors.

Accuracy 0.3% Input Resistance 20K Ohms Typ

A few applications may include monitoring and control of ● flow ● temperature ● humidity ● wind speed ● wind direction ● light intensity ● pressure ● RPM ● storage oscilloscope ● soil moisture and many more.

We also manufacture a 16 channel digital input/output board for control applications.

PRICE $129.00

NEW

IIe Only: 80 Column, 64K RAM Card

- Expand your Apple IIe to 128K memory.
- Provides an 80 column text display.
- TOTALLY compatible with ALL Apple software and languages, there are NO exceptions.
- Automatically expands VisiCalc to 95K storage. In 80 columns!
- COMPLETE documentation included. (We don't make you refer to the Apple manual as others do.)
- Uses the same commands as the Apple 80 column board.
- Incorporates the latest high speed, low power I.C. technologies.
- Plugs into the Apple IIe expansion slot.
- Simply the best expansion card for your Apple IIe at any price, offering you phenomenal performance at a very nominal price.

PRICE $149.00

Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products work in APPLE IIe, II and II+. (Except 80 column card.)

Applied Engineering's products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle two year warranty.

All Orders Shipped Same Day
Texas Residents Add 5% Sales Tax
Add $10.00 If Outside U.S.A.
Dealer Inquiries Welcome

Send Check or Money Order to:
APPLIED ENGINEERING
P.O. Box 470301
Dallas, TX 75247

Call (214) 492-2027
7am to 11pm 7 days a week
MasterCard, Visa & C.O.D. Welcome

SF APPLE CORE
1515 Sloat Blvd., Suite 2
San Francisco, CA. 94132

TIME VALUE

Bulk Rate
U.S. Postage
PAID
Permit #85
El Cerrito, CA 94530

TO:

3121AUG83
DOUGLAS STORKOVICH
921 PINTO CT
WALNUT CREEK CA 94596